NAVYA KHURANA

B.S. in Computer Science + Math Minor at UMD

A highly motivated learner, interested in combining computation & AI with other fields to develop unique solutions to complex problems.

US Citizen | +1 425-469-5622 | linkedin.com/in/navya-khurana
nkhurana@umd.edu | github.com/nkhur | College Park, MD

EDUCATION

University of Maryland, College-Park *2023-2026*

- President's Scholarship Awardee
- Dean's List
- GPA: 3.87
- Relevant Coursework: Introduction to Data Science (ongoing), Algorithms, Organization of Programming Languages (OCaml, Rust), Data Structures, Computer Systems (C, Assembly, etc.), Object-oriented Programming in Java, Multivariable Calculus (ongoing), Applied Harmonic Analysis (Signal Processing) (ongoing), Linear Algebra, Discrete Math, Calculus II, Introduction to Psychology, Cellular and Molecular Biology

The Shri Ram School, Gurgaon, India *Middle & High School: 2014-2023*

- Ranked First in Computer Science
- AP Scholar
- Percentage: 96%

TECHNICAL SKILLS

Languages: JavaScript / Typescript, Java, Python, Rust, C, R, OCaml, AVR Assembly
Frameworks: React, Node.js, Next.js, JUnit, Streamlit, PyQt6, Flask
Dev Tools: MongoDB, LLMs (OpenAI, Gemini), SQL, Git, Figma, Canva, MATLAB, RStudio.
Libraries: Pandas, Numpy, Matplotlib, Seaborn

WORK EXPERIENCE

Undergraduate Researcher @ Digital Engagement Lab, UMD *March 2025 - Present*

Python (Pandas, Numpy), TypeScript | *With* **Professor Ronald Yaros, College of Journalism @ UMD**

- Finding data using YouTube API for audience engagement with news channels + using NLP to drive insights (eg. averaging the sentiment analysis on comments, detecting curiosity-creating keywords in the title).
- Implementing tools to observe user interaction with the "New View News" site - under development as a more interactive and engaging news format.

Software Engineer Intern @ Passionfruit, New York *July 2024 - September 2024*

Node.js, Next.js, Python, MongoDB

- Created AI-driven and LLM-powered tools and solutions, to help create optimized SEO strategies and content generation for various clients. Projects:
- SEO Content Ideas Generator: Data from Google's popular questions + top blogs structure (word limit, # of links, # of images, etc.) used in LLM prompt.
- Topical Authority Scorer: Assessing SEO relevance for inputted content and a chosen topic. Extracted keywords from content text using YAKE and BERT. Scoring keywords, headings, metatags, & url via text-embeddings + cosine-similiarity and LLM APIs.
- Chatbot for Suggested Actions: Based on performance data of websites (fetching from mongoDB). Used linear regression to analyze metric changes over specified duration for each specific SEO task.
- Content Editor: With real-time keyword analysis + autosave, rewrite, and download functionality.

Software Engineer @ Shah Lab, UMD *February 2024 - September 2024*

Python | *With* **Professor Sahil Shah, Computer Engineering Department @ UMD**

- Project: Creating a neural system that sends & records neural signals sent to brain.
- Designed + developed GUI used to send, clean, and record neural data using bit manipulation; neural interface connected via RESTful API (Module: Flask). Handled any firmware requirements as needed.
- Generating .coe files from inputted SPI Commands, as per Intan RHS 2000 Board requirements.
- Analyzed C++ OpenEphys repositories to integrate functions for specific board requirements + select plugins to connect GUI with OpenEphys

PROJECTS

Professor Rating Predictor *2025*

Python (Pandas, scikit-learn)

- Built 3 ML models to predict a professor's rating using the PlanetTerp API (similar to ratemyprofessor for UMD students) without using the actual rating as the feature. Tested to determine the best model, presented in a slide deck.

MicroCaml Language Development *2024*

OCaml

- Implemented custom lexer, parser, and interpreter for MicroCaml, a dynamically-typed language inspired by OCaml.
- Developed optimizer that simplifies ASTs created by the parser to increase runtime efficiency.
- Built a type checker to ensure type safety and prevent runtime errors.
- Added OCaml functional programming principles: lexical scoping, recursive functions, let bindings, complex control flow constructs.

Shell Implementation *2024*

C | *UNIX Commands, Pipes, System I/O, Process Control*

- Supports certain commands: &&, pipes, input output redirection, and subshell.
- Pipes and subshells handled using a tree structure (code making the tree from input given).
- Use of child processes to handle specific commands

Finding All Possible Solutions of a Maze *2023*

Java | *Graphs*

- Each (x,y) in the maze, represented as an instance of class Juncture, is a vertex of the graph.
- Represented a maze as a weighted graph using ArrayLists, storing instances of static inner class Vertex. Ensuring edges only added if maze positions not separated by walls.
- Finding the solution(s) to the maze via Breath-First and Depth-First Search. Implemented Djikstra's Algorithm to find the most optimal solution. (GUI Provided)

SafeSites *Technica Hackathon, 2023*

Python | *Library: Streamlit (Front-End)*

- Goal: To ensure users can find safe properties for their stay via AirBnB.
- Finds data about the crime rating of the locality a user is looking at. Uses other factors (number of children, pets, etc.) needed to find the optimal AirBnB, displays properties in order of safety rating.

EXTRA-CURRICULAR INITIATIVES

Head of Journalism, Aravali Model UN *2022*
Media & Writing Intern, CovidTales.org *2022*

INTERESTS

Reading, Music, Watching Formula One Racing